OVAL Board Meeting (1/12/2009)

Attendees

Jonathan Baker – MITRE
Andrew Buttner – MITRE
Bryan Worrell – MITRE
Margie Zuk – MITRE
Melissa Albanese – DoD
Scott Armstrong – Gideon Technologies, Inc.
Carl Banzhof - McAfee
Scott Carpenter – Secure Elements, Inc.
Anton Chuvakin – Qualys, Inc.
Nick Connor – Assuria Limited
Jonathan Frazier – Gideon Technologies, Inc.
Jay Graver - nCircle Network Security, Inc.
Morey Haber – eEye Digital Security
Rob Hollis – ThreatGuard, Inc.
Kent Landfield – McAfee
Tim Keanini - nCircle Network Security, Inc.
Alex Quilter, Hewlett Packard
Stephen Quinn - National Institute of Standards and Technology (NIST)
David Waltermire - Booz Allen Hamilton

Agenda

- Welcome
- Status Update
    - OVAL Language Release Process
    - OVAL Adoption
    - OVAL Language
    - OVAL Interpreter
    - OVAL Repository
- Board Refresh
- OVAL Compatibility/Adoption Update
- Versioning Process
    - Allowable Minor Version Impact
    - Deprecation Process Improvement
    - Allowance for Breaking Backward Compatibility
- Language Expansion vs. Major Revision

- Release Roadmap
- Questions/Concerns

Meeting Summary

Welcome

After introductions the group was welcomed to the 2009 1st quarter OVAL Board Meeting. Attendance was very good for this meeting and everyone was excited to see the renewed commitment to the OVAL Board. Three new board members were introduced to the group:

- Scott Armstrong – Gideon Technologies, Inc.
- Jonathan Frazier – Gideon Technologies, Inc.
- Anton Chuvakin – Qualys, Inc.

Status Update

A brief status update of the OVAL project as a whole was delivered. The following items were covered:

- OVAL Language Release Process
  There is an active discussion going on the OVAL Board list on evolving and maturing the current release process. We will spend much of our meeting today on this topic. Out aim is to better define the release process and then develop a long-term release roadmap for OVAL. Thanks to all for the participation in this discussion.

- OVAL Language
  Version 5.6 had been planned for late January. We have decided to put that on hold until the release process discussion is completed. There are several minor items in queue for version 5.6. We should get started on 5.6 as soon as the release process discussion is finished.

- OVAL Interpreter
  Work is being done to import Linux support in an effort to enable the SCAP Validation Program to test on Linux systems. Bryan Worrell has added a run level probe and will be working on the text file content test next.

- OVAL Repository
  The OVAL Repository continues to be current for all windows advisories. There are active Sun, HPUX, and AIX content contributions as well.

Board Refresh

Please have a look at the responsibilities overview posted on the web site (http://oval.mitre.org/community/board/index.html). We greatly value board members' engagement and participation. In an effort to keep the board actively engaged we will reach out to vendors from whom we haven't heard in a while and we may bring on some new vendors.

OVAL Compatibility/Adoption Update

NIST is working to further mature SCAP and the SCAP Validation program. To simplify the current SCAP DTR, a new DTR will be developed for OVAL. Infrastructure improvements to provide better access to the NIST Validation data are on the way as well. Both of these changes have delayed the completion of the transition to NIST run Validation until April 1st.

MITRE will be developing new declarations, questionnaires, and technical use case documentation for the new adoption program over the next few months.

Versioning Process

After the October 2008 OVAL Board meeting MITRE developed an OVAL Language Release Roadmap and shared it with NIST and the OVAL Board. The roadmap proposal lead to a great deal of discussion about the current OVAL Language versioning process. Several board members have expressed strong concerns about the notion of a release breaking backward compatibility and have raised larger questions about the versioning process that is in place for the OVAL Language.

Currently, major versions are defined to be revisions that can break backward compatibility. There has been a fair amount of discussion on this topic. Several board members have expressed strong concerns with breaking backward compatibility. Based on the conversation it seems as though we should continue to allow a major version to break backward compatibility as long as we have a well defined deprecation process that allows for a great deal of lead time before a change is made. The hope is that the community will have ample time to respond and raise concerns about specific changes.

As a reminder, ad hoc conference calls for any of these items can be arranged. We can also leverage mutual presence at conferences or make other arrangements to meet in person as needed.

Stephen Quinn gave the group an overview of NIST's current work to develop an official specification for SCAP and a FIPS based on that specification. NIST is also working to develop its own release roadmap for SCAP versions.

Allowance for Breaking Backward Compatibility

Several board members have expressed strong concerns with breaking backward compatibility. Based on the conversation it seems as though we should continue to allow a major version to break backward compatibility as long as we have a well defined deprecation process that allows for a great deal of lead time before a change is made that ensures that the community has ample time to respond and raise concerns about specific changes.

Jonathan Baker: Currently, the only way we can break backward compatibility is with a new major version. This is making a change to a new version of the language which will prevent a document written against an older version from validating against the newer version.

One problem with the current process is that we can go from 5.x to 6.0 and drop support for some feature and effectively force vendors to update their content.

This is something that we probably need to avoid and consider preventing by redefining our versioning process.

As the discussion has evolved on the board list I was under the impression that people were okay with allowing a major version to break backward compatibility as long as a deprecation process were in place that is well-documented, provides transparency to the vendors community, and allows for a with a lot of lead time before a change is made.

Question: What will cause something to be deprecated? Lack of use?

Response: There is any number of reasons; it's not used, broken, mistakenly added.

David Waltermire: If there is going to be a published policy on it, I would like to see it weighted toward content that currently exists as published. From a workflow perspective you have to know it is deprecated well ahead of its being removed. At NIST we use a least version principle. Basically, we make every effort to keep our published content on the least version that supports our needs. This helps ensure the maximum number of vendors can utilize out content. Additionally, why deprecate anything if you aren't going to get rid of it?

Question: What will deprecation do to operationally deployed systems? Why invalidate it if it works?

Response: Part of the deprecation policy should be defining how deprecated items are processed by content consumers.

Deprecation Process Improvement

In response the board discussion we plan to develop a "Deprecation Policy" page on the OVAL web site. The policy will be developed and finalized over the OVAL Developer List.

MITRE has done a survey of deprecation processes and practices in place in other languages, and is working on an updated deprecation policy based on the findings. The intent of the updated deprecation policy is to ensure that people will understand with each release:

- what is a candidate for deprecation
- which, if any at all, previously deprecated items are candidates for removal from the language
- ensure complete openness and transparency in the process of evolving the language
- clearly define the end state of a deprecated item

After an initial review by the board the draft deprecation policy will be distributed to the OVAL Developer List for review. Once completed the new deprecation policy will be published on the OVAL web site.

Comment:	Data maintainability: we have to make sure the process can be supported by the tool in question. We need a mandatory support period; once the support period is through doesn't mean the content is invalid.
Comment:	The process should be to ensure that deprecated items will be identified as such and will not be removed if still in use.
Comment:	How can you determine what language constructs are being used within the community? We have little visibility into what is in use and what is not.
Response:	The point is that we want vendors to have input into any given deprecated constructs.
Question:	Once it is deprecated can it be undone? Getting a tag removed if something is still in use and the user is not aware of it? This will have to be made very public so that everyone knows. Also need a machine readable component to the content.
Response:	We've thought of using schema constructs to mark items as deprecated. We would like to maintain a list of what has been deprecated and what is slated for deprecation to ensure that all users have a clear understanding of the changes that are planned and can raise concerns if needed.
Kent Landfield:	Is a face-to-face meeting in order? Maybe we need a high bandwidth discussion on this. Let's think about a schedule for a meeting in person or a focused telecon.
Dave Waltermire:	We can do this on site at NIST, if everyone would like.
Jonathan Baker:	I will look into setting something up.

Allowable Minor Version Impact

Should we stick to the documented scope of a minor release and allow any backward compatible changes? A good example is the change proposed to the object structure for version 6. This change would not break backward compatibility, but would add in a new construct that all tool vendors would need to implement to support the new minor version. Here is a link to the discussion on the proposed changes to the object structure:

http://n2.nabble.com/Choice-inside-an-OVAL-Object-tp1485589p1485589.html

Question:	Is there a concern about adding impactful features to a minor revision?
Kent Landfield:	Yes, definitely. This sort of new feature addition should be reserved for release in a major version.

Question: How do I know which tools have been validated to test the new features that have been introduced?

Jonathan Baker: For now you would have to understand what version of OVAL the tool was validated against. By adding a new feature the version of OVAL would be incremented. If the tool you are using was validated on an older version it would be safe to assume that it did not support the new feature.

Language Expansion vs. Major Revision

Along with all the discussion about versioning process, there has also been discussion about shifting focus from developing a new major revision and moving it toward expanding OVAL to support new platforms. MITRE is open to expanding OVAL's platform support but requires community-leveraged expertise and guidance in order to help develop new platform schemas. The board was asked if this shift in focus was appropriate.

In response there were arguments for working on expanding to new platforms and against expanding before we release the next major version. The risk in expanding to new platforms before we move to a new major version is that vendors will become further attached to version 5.x of OVAL and it will be more difficult to move to a new major version in the future. However, if we do expand to new platforms before we develop the next major version of OVAL we might gain insight into new capabilities that should be supported in the next major version.

To summarize the Board's opinion, MITRE should make sure that development of a new major version does not cause expansion to stop. We need to advance down both paths (development and expansion) to continue the positive growth of OVAL.

Release Roadmap

The versioning process conversation started with an attempt to define a release roadmap. There are currently several minor changes in the queue for the next minor release. Once the versioning process conversations have wrapped up we will propose a new release roadmap. Most likely with a next minor version released around the end of March.

Questions/Concerns

- OVAL for vendors with NAC solutions
  Recently within MITRE there have been questions about using OVAL in NAC solutions. Are any vendors currently doing this or considering it?
  Kent Landfield: We are using it for NAC but we definitely have some limitations.
  Morey Haber: We are doing the same with partners.

Actions

MITRE will look to arrange additional teleconferences and/or face to face meetings to advance the version process discussion.